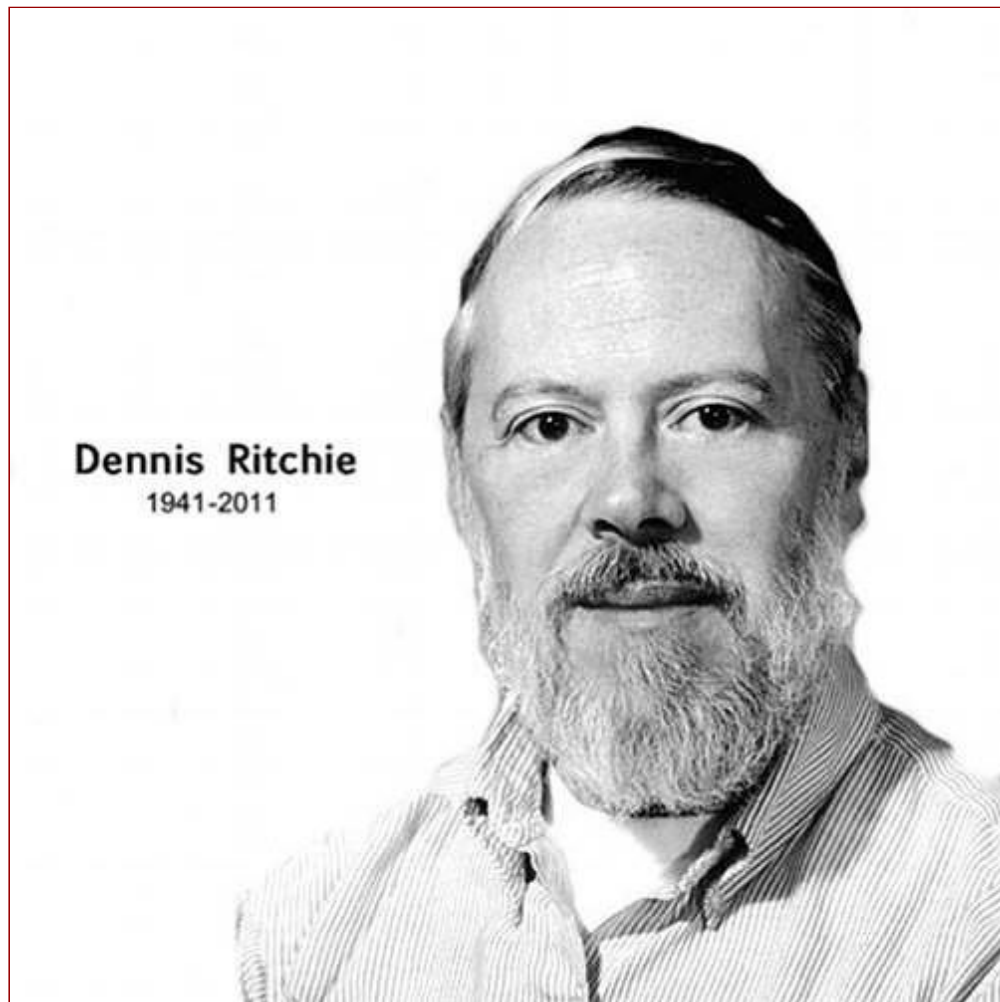


CS3012 : Assignment 3

Biography of an influential software Engineer

Cormac O'Dwyer - 15325054

Dennis Ritchie



Early Life

Dennis Ritchie was born on the 9th of September, 1941 in Bronxville New York. He was born to Alistair Ritchie who was a switching systems engineer working in Bell Laboratories where Dennis himself will work after finishing his college education, and Jean McGee Ritchie. Dennis performed well during his high school education and later went on to study both Physics and Applied Mathematics at Harvard University. During his studies at Harvard it is said that he went to a lecture about the computer system that was in Harvard at the time, a Univac 1 and was enthralled by how it functioned.

Outside of his studies Dennis began to learn more about computers and how they operated. At the time computer science was not a degree but computer labs were desperately trying to find anyone with potential to help with their work. This gave him an opportunity and managed to get himself a job working at the Massachusetts Institute of Technology, MIT for a number of years.

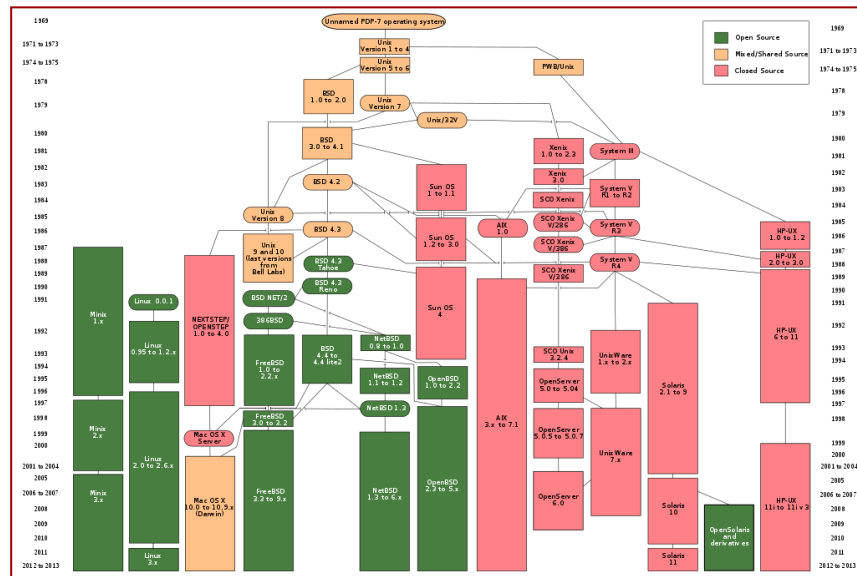
Unix :

After finishing his college education Ritchie decided to pursue a career in computing rather than physics or maths and decided to work at Bell Labs Computing Sciences Research Center, where his father had been working. During that time he was working on a project that was being collaborated by Bell Labs , MIT and General Electric to develop a 'time-sharing' system. 'Time-sharing' is the sharing of computer resources among many differing users via multiprogramming and multitasking simultaneously. The project was called Multiplexed Information and Computing Service, or 'Multics' for short. But in 1968 with Bell Labs being unsatisfied with the progress ultimately pulled out of the project.

During this time Dennis along with his colleague, Kenneth Thompson became interested with the growing popularity of the so called minicomputer. Computers during this time were primarily specified for a number of small procedures and usually took up a large amount of space meaning computers were not a widespread commercial item and were primarily used by businesses to speed up certain tasks. That started to change with the release of smaller, more portable machines. This rise in use of minicomputers highlighted the need for a standard, easy to use operating system that allows for a more feasible interaction amongst various computers. Ritchie and Thompson decided to create their own OS that could fill this criteria drawing heavily from their experience working on the Multics project.

This operating system would become Unix. Operating systems at the time were very complex and not intuitive to use meaning only specialists in the computing field could navigate them without issue , this changed with Unix as it was comparatively cheap, easy to use and could be run on nearly any computer.

The lasting impact Unix has had is staggering , most operating systems can be categorized into two families. On one side you have Microsoft's Windows NT-based OS, with almost everything else being able to trace its heritage back to Unix. You can still see the Unix philosophy influence many of the popular OS of today like linux or Mac OS, from their kernel and File Systems.



C:

The very first implementation of Unix was written in assembly language but very quickly it became apparent that it was simply too taxing to write an entire OS purely in assembly. Originally the team that developed Unix wanted to rewrite the system using a language known as 'B-language' which was Kenneth Thompson's simplification of the 'BCPL' language, but due to some limitations of the B language, like the inability of byte addressing lead Dennis Ritchie to develop what came to be known as the C programming language.

C uses miniscule syntax with few instructions but it very structured and modular which meant it was very easy to use in different computers. This allowed for programmers to copy and reuse blocks of C code and it would run just fine on their own machine without issue. These blocks would later be put into libraries allowing programmers to not have to start from scratch at every project meaning a quicker and easier to time implement code.

In the early 1970's the Unix kernel was rewritten using C, after C implemented its 'struct' type making the language very powerful. With the popularity of Unix spreading in some form to nearly most computers in the 1980's meant that the C programming language was nearly mandatory to learn for every programmer. The effect of this is still widespread to this day with C still being regarded as a basic skill of most programmers, the use of C as a language is still hugely widespread, the influence C has had on many other programming languages can be seen such as java, C#, C++ , python and many more.

To this day the touch of Dennis Ritchie can be felt with the ever prevalent influence of both Unix and the rise of C along with it.

